SquawkBox for PS1.3a
User Manual and Tutorial

Kees Leune (kees@leune.org)
Jeroen Hoppenbrouwers (hoppie@home.nl)

February 8, 2003

Contents

Acknowledgments

This program could not have been written without the help of a number of people. First and foremost, Hardy Heinlin, who wrote 747-400 Precision Simulator and has been the driving force behind most of the developments around this piece of software.

Second, the people behind VATSIM and IVAO, the organizations that enable global air traffic coverage over the Internet. Joe Jurecka and Jason Grooms, the authors of SquawkBox for MSFS95/98/2000 and ProController, both provided valuable input for our project. Phillip Dale and John Eisenhour who run(ned) the SATSERV server network cooperated in a great way during the testing phase of the program and helped us getting everything just right.

Third, Ivan Ngeow and Dušan Lacko for their implementation of the link with PS1.2c and all the fights with a certain operating system that resulted from this. Without them, SB747 would not have run on 1.2c at all.

Fourth, a group of dedicated beta testers that made sure that our program did not contain too many goofs and insured usability with PS1 and the rest of the VATSIM/IVAO community. Their comments and suggestions were highly valued.

Lastly, the whole active PS1 community that meets regularly at the PS1 Message Board[1] and that has provided us with continuing and very constructive support.

Kees Leune
Jeroen Hoppenbrouwers

[1] `http://aerowinx.com`

Chapter 1

Introduction

SquawkBox for PS1 is the pilot/plane side of a larger system. This larger system consists of a cluster of Unix server machines on the Internet that coordinates the message traffic between all connected pilots/planes and air traffic controllers/ground stations. In this manual, we will call this system *the network*.

There are several organisations providing the same service with essentially the same software. As long as the protocol is the same, SB747 can connect to all these networks. VATSIM and IVAO are the biggest networks at the moment.

For quite a time, virtual ATC was almost exclusively accessed by pilots flying Microsoft Flight Simulator, using the freeware SquawkBox program by Joe Jurecka. With the introduction of SquawkBox for PS1, pilots now can get their hands on the controls of the only current true heavy jet simulator and join the sightseeing crowds in the skies all over the world.

1.1 What You Can Expect

SB747, like the original SquawkBox program, must be used on top of a flight simulator, in this case the Aerowinx 747-400 Precision Simulator. The SB747 program will let you connect to the server cluster, log in as a licensed pilot, and contact all active ATC stations from clearance delivery via ground control and tower to departure and center, all the way to the arrival airport. You use a text-based interface (message lines) on the PS1 flight deck to communicate with the controllers, or a live voice system called *RogerWilco*. SB747 contains most bells and whistles of the original SquawkBox, such as private communications, METAR and ATIS reception facilities, and many features that make the system nice to use. You even have the same weather in the simulator as there is on the same spot in the real world.

People using MSFS with SquawkBox *will see you through their windshields*. If they have set up their systems correctly, they will even see you as a Boeing 744, with the proper company livery.

Combined with PS1's outstanding simulation of a 744's systems, SB747 provides the finishing touch in simulated IFR flying.

1.2 What You Can Not Expect

The most salient aspect of a crowded flight environment will not be present: you will not see other aircraft in stunning graphical detail. This is due to the limited graphical capabilities of the current major version of Precision Simulator and cannot easily be improved. In practice, this is not a severe limitation, except when taxiing and waiting in line for takeoff clearance. All other phases of flight are strictly IFR.

However, several programs are available to extend the current PS1 software with a direct link to MSFS so that you can use this graphical workhorse for its natural application – looking out of the window – while still 'flying' PS1 on another computer.[1] [2] Using this system, you *can* see the other planes, and most of the weather, too!

Likewise, the TCAS aboard the PS1 744 is not linked up to the virtual environment. However, linking up TCAS is far less complex than providing the outside scenery. An external program already provides some of the required functionality, see Section 7.6. Other traffic using MSFS will see you on their TCAS, though, just as with outside scenery. SB747 provides the scenery generators described in the previous paragraph with the data required for traffic display, but as of this writing, neither of them supports this feature yet.

Markus Vitzetum's external instrument panels[3] include the traffic information of SB747 and even calculate proper TCAS warnings.

[1] http://www.jcavanagh.flyer.co.uk/si/alpha.htm
[2] http://mypage.bluewin.ch/Visual744/
[3] http://www.displays.glideslope.de/

Chapter 2

Installation

To install SB747, just create a suitable directory somewhere (we suggest calling it `sb747`) and unpack the `.ZIP` file in this directory with a suitable unzipper, such as WinZip, PKUNZIP, or UNZIP. The result will be three additional directories inside your directory, `bin`, `etc`, and `lib`. Make sure you use the 'create directories' option of your unzipper (-d with PKUNZIP, *Use Folder Names* with WinZip)!

You now have installed SB747. If you are in High Anxiety Mode with the Broker already installed and running, and just want to get going, you are cleared to proceed directly to Chapter 3.

2.1 Files

This section describes the purpose of the files that make up SB747.

In the `bin` directory you will find a single, rather large (3Mb) executable called `sb747.exe`, in `lib` one file called `priority.dll`, and in `etc` four files called `aircraft.cfg`, `hosts.txt`, `icaopos.txt`, and `expand.txt`. These small files are plain ASCII lists of network servers, supported aircraft liveries, ICAO weather stations, and phrase shortcuts, respectively.

You only need to change the `hosts.txt` file if there is a change in the server cluster. Newer releases of SB747 will include all required modifications, but you can make them yourself as well. Some firewall setups also might require changes in this file. In case you fly on more than one network, you can add your pilot ID and network password after the server lines, using colons (':') as separators just as the rest of the line. If you do this, selecting a new server will also select the correct pilot ID and password.

The file `aircraft.cfg` is a plain copy from the CSL plane model library[1] for the 'original' SquawkBox by Joe Jurecka and others. It contains all plane models that correctly show up in the Microsoft flight simulator. When a new CSL library is released, you can replace this file by the new one. SB747 will correctly pick up the available 744 model liveries from this file.

The `icaopos.txt` file contains over 6000 ICAO stations and their positions in latitude and longitude, plus elevation data. SB747 uses this list

[1] `http://avsim.com/mike/sb_pc/csl/`

to look for the nearest weather station while you are in flight and for some altitude corrections to winds, clouds, and temperatures. You may edit this file as needed.

The `expand.txt` file is described in more detail in Section 7.5.

When you run the `sb747.exe` file, it will create `etc/sb747.ini`, in which your personal preferences will be stored. For debugging purposes, `etc/debug.log` will also be created. This file may be thrown away at any time, but in case of problems, it might contain valuable tracing information that the SB747 authors might find useful.

There are no other files involved in the SquawkBox system. If you delete all files and the directories, every trace of the program is gone from your machine. If you want to move the program, just move all files together, as long as you maintain the structure with the three subdirectories.

The reason why you get a small directory tree instead of just a flat structure has to do with the development of SB747. Full source code releases, when available, will be able to gracefully coexist with the current wrapped version in the same directory tree.

2.2 The 747 Broker

SB747 just by itself cannot work with PS1. You need one additional program for this, called *the Broker*. It can be gotten from the same Web site as you got SB747 from.[2] The Broker comes with its own documentation. Please make sure it works correctly before you try out SB747.

2.3 Starting SquawkBox for PS1

There are various ways of starting SB747. You can start it directly from the `Start/Run` menu of Windows. This is convenient for a first try, but not suitable for permanent usage.

You can also start SB747 from a DOS window, which has the same drawbacks as from the `Start/Run` menu. However, you can write a *batch file* that includes commands to start both SB747 and friends plus PS1 together, or does even more things such as setting the volume of your sound card and download yoke commands.

Lastly, you can start the program through an icon or shortcut that you make. This probably is the most handy way on Windows, especially if you also write a startup batch file which fires up PS1 and the associated helper programs on top. Putting the program somewhere in the `Programs` menu is a good alternative.

All ways of starting are technically equivalent. You do not need to switch to the directory where the executable is located, the program will find out where it lives and from there it searches for the other required files.

[2] `http://www.hoppie.nl/`

2.3.1 Recommended Start Order

There is a preferred order to start all programs, although they do not depend on this order. Generally, it causes the least hiccups and error messages when you start the Broker first, then SB747, then (if you want to) RogerWilco, then PS1 which will take you to full screen mode. You need to start RogerWilco before PS1 so that the first gets the sound card. With some sound cards, like the SoundBlaster Live! series, this is not any longer required, and you can let SB747 start RogerWilco when it is tuned for the first time.

In case one of the programs crashes, just restart it. All programs are designed to re-establish communication with each other as soon as they reappear on your system. In case SB747 itself crashes, you need to make the server connection again manually.

Chapter 3

Getting Started

SquawkBox for PS1 has one main window with very few controls, since most work is being done from the PS1 flight deck (as it should). You have a `File` and a `Help` menu, a log area, a VHF entry line, a byte rate indicator, two activity indicators, and a three-way toggle switch. The menus are straightforward, but most items in the `File` menu will only be meaningful when a server connection has been opened successfully.

The log area shows a growing list (maximally 500 lines) of events that might be of interest to you. This list is a mixture of system messages and

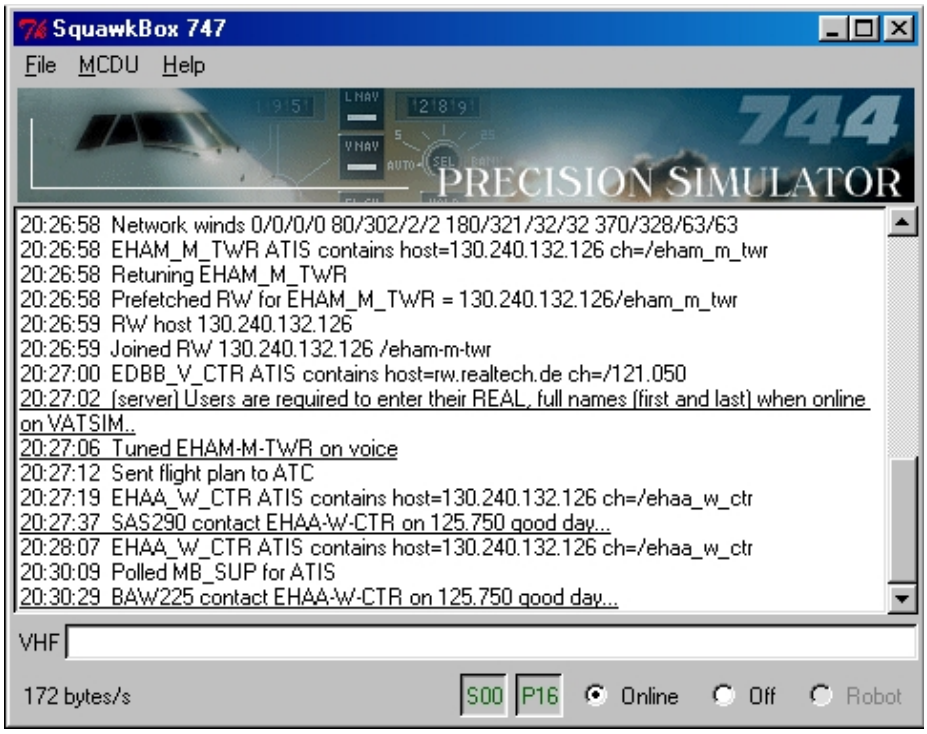

Figure 3.1: The SquawkBox for PS1 Main Window

texts that are transmitted by the network controllers and pilots. Everything that is also sent to the PS1 outside view window appears in **boldface**. Generally you only look here when you are puzzled by something, or to get confirmation that a certain action completed successfully. This area is *not* the main interface to the controller, because you will use PS1 itself for that purpose. Should you find yourself in a position where you would want to interact with SB747 directly though, then you can type your messages and commands in the VHF entry line and press Enter when done. The effect is the same as from the PS1 flight deck.

Next to the three-way toggle you find a byte rate display and two activity indicators. If you are online and receiving data, the byte rate display simply tells you how many bytes per second flow in. The indicator marked with an 'S' is for Server, the 'P' is for PS1. Each time a packet is received from the respective transmitter, the indicator turns green and increases (up to 99). If no packet has been received for a while, the indicator turns red. This does not necessarily mean that you have a problem, but it is worth investigating the cause. If the 'S' indicator turns red, there has been a suspiciously long period without any sign of life from the server. SB747 will detect this and also warn you by a VHF message. If nothing happens any more, cycle the connection off and on or switch to another server. In general, try to keep both indicators green during a live session.

The three-way toggle switch 'Online, Off, Robot' lets you connect to the network, disconnect, or connect to the stand-alone (offline) ATC robot.

The program can generate a variety of other windows. These windows act as child windows of the main window: when you minimize the main window, the child windows will disappear as well. All windows can be moved and resized, and most of them will remember what you did to them.

3.1 Setting Up SquawkBox

The first thing you should do is to open the `File/Setup` menu and fill in the dialog box that pops up. It is divided into two parts, one for the server connection, one for SB747 local settings.

If you do not yet have a (virtual) pilot or controller license, apply for one at your network of choice. Before you select a server to connect to, you should check the same Web site to see which servers currently are online, and also to look for NOTAMs which might influence your flight. At the same time you can note the controllers which are active, which might influence your flight.

You can easily switch servers in mid-flight by disconnecting, changing the server in the Setup box, and reconnecting. This will take you off a controller's screen for a few seconds only. Please fill in your *real name* at all times. You may add the nearest airport ICAO code to your name to give a hint about where you live. Pilot ID and password are handed out by your network and are *strictly personal*.

In the SquawkBox Configuration part of the dialog box, you must fill in the host IP address where the Broker can be found. In case the Broker runs on the same machine as SB747, leave the default of 'localhost'.

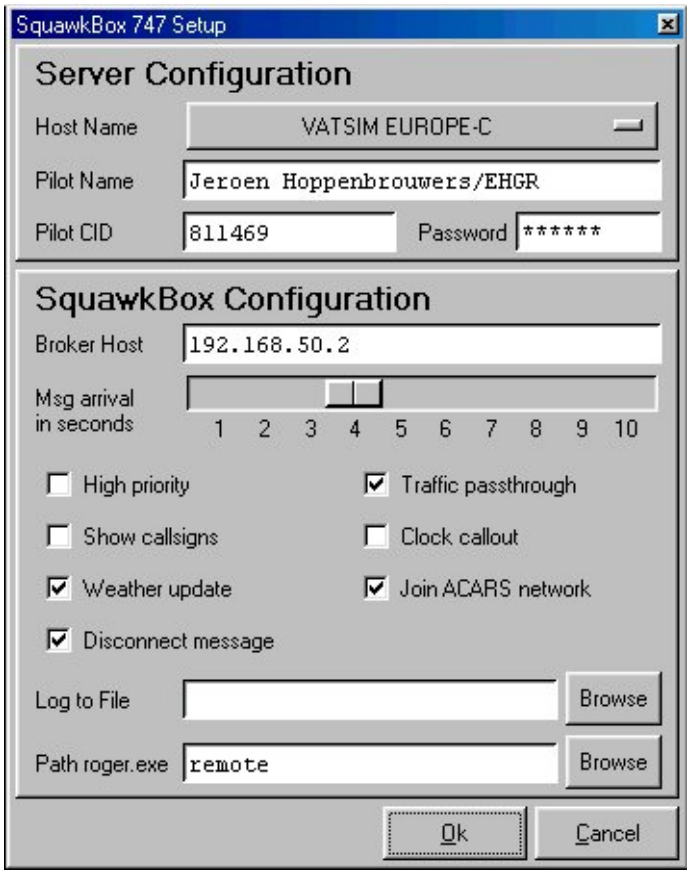

Figure 3.2: The Setup Dialog Window

The slider sets the minimum number of seconds in between messages in PS1. When you do not have enough time to read the messages, increase this number.

On slower systems, the 'High Priority' setting might improve performance and decrease network lag. This switch will only take effect the *next* time you start SB747.

'Show Callsigns' can be handy in case you lose track of who is saying what to who. In reality, radio messages of course are not automatically tagged with the identity of the transmitter – that's why there is such a strict protocol of callsign use! But virtual pilots and sometimes even controllers tend to get lazy about this, because the SquawkBox for MSFS always displays the originating callsigns automatically. With this switch you can select whether or not you want the extra callsign display.

Most times you would want automatic continuous weather updates, reason to check the box 'Weather Update'. This synchronises the weather in PS1 with the weather such as exists in the virtual world. This weather is more or less synchronised with the real weather in the real world.[1] If you

[1]The real-world weather stations submit reports every hour. These reports are collected by

leave this switch off, SB747 will not touch any weather setting, and another program such as SimWX[2] can take over.

With 'Disconnect Message' you can select whether or not you want a notification sent to you by ACARS if the network goes down. You will get a message in the outside view window in any case.

The 'Traffic Passthrough' setting should only be switched on if your system is powerful enough and if you want it. With this switch off, you do not get any pilots in the 'Pilots in Range' box and if you have any TCAS-capable program connected to the Broker, it won't receive traffic information. The upside is that you will *significantly* reduce the CPU load that SB747 causes, especially when there is a lot of traffic out there.

In case you fly with your PS1 board clock set to real time, you will want to select 'clock callout,' so that your F/O warns you when the board clock deviates significantly from real (Zulu) time.

'Join ACARS network' enables the external ACARS functionality that you might want to use if you fly for a virtual airline. See Chapter 8 for an in-depth description of the external ACARS functionality.

If you want to log everything that appears in the Log Window to a file, fill in the file name in the 'Log to File' box. Leaving it empty means no logging takes place.

Should you desire to have RogerWilco[3] integrated with SB747, you must fill in the full path to the `roger.exe` executable, too. Include the file name itself in the path. In case you use the separately available remote RogerWilco tuner,[4] so that you can run RogerWilco on another machine as SB747, fill in `remote`.

Your settings are saved in the file `etc/sb747.ini` and will remain permanent for as long as this file exists, or until you change them using the `File/Setup` dialog box again. There is quite a chance that you will not need to touch them any more after you have set them once.

3.2 Filing a Flight Plan

In order to either connect to the network or to run the ATC robot, you *must* file a flight plan. Open the Flight Plan dialog with `File/Flight Plan`. Most fields are to be treated as if you were filling in a real paper flight plan; they will appear exactly as you fill them in to the air traffic controller(s), and they are not validated. Take care in selecting your call sign, because you will not be able to change it during your online session. All other fields can be changed while you are online, so you can change your mind or divert simply by changing and then resubmitting your flight plan. Not filling in a field may lead to the network server rejecting your flight plan, so please fill them all in, even if it is with zeroes.

---

some institutions such as the NOAA. The virtual air traffic networks download this information about every 90 minutes, so you might see a delay of a couple of hours compared to the real weather.

[2] `http://www.weather.glideslope.de/`

[3] `http://rogerwilco.gamespy.com`

[4] `http://www.hoppie.nl/rwd/`

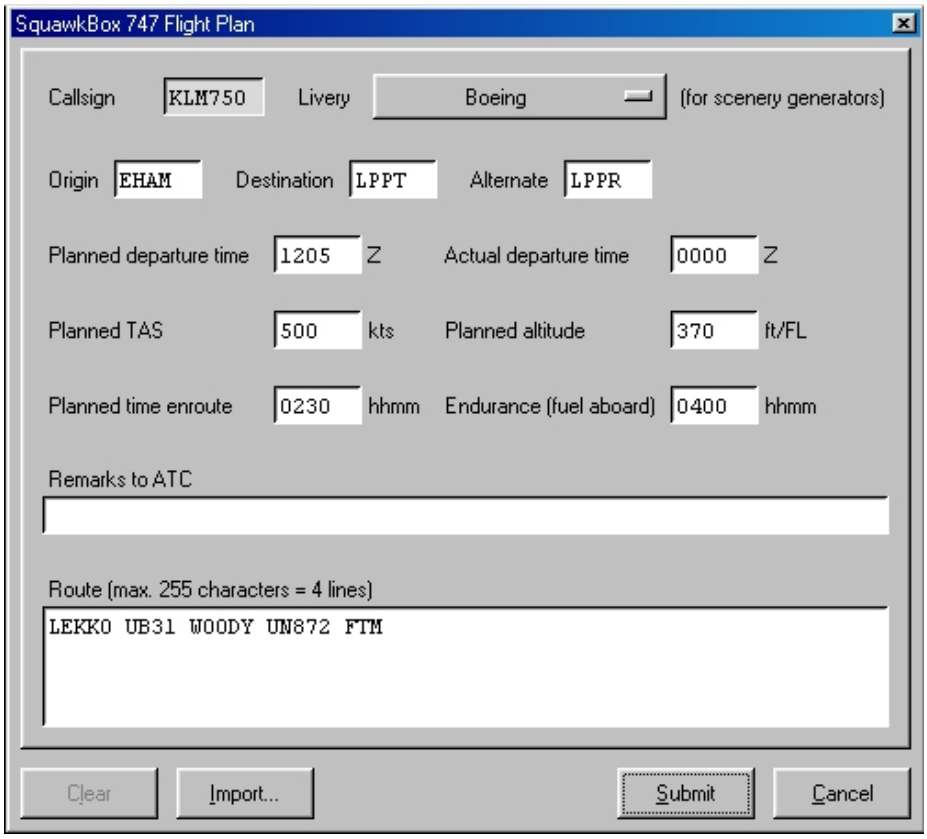

Figure 3.3: The Flight Plan Dialog Window

You must select a callsign that is unique at that moment for the whole network. An appropriate livery selection is up to you; select 'Boeing' for a no-frills standard Boeing 747-400 factory look.

Origin, destination, and alternate airports (using four-letter ICAO codes) are straightforward, as are most other items. Your actual departure time obviously can be filled in only after you left the gate. Endurance is the estimated time you can stay airborne given your fuel load and burnoff rate. Use the Remarks line for generic remarks about your intentions; they will appear on the controller's flight strip next to your route. You can decide for yourself how detailed your route description will be. Note that there is a maximum of 300 characters in the route field.

It is a good idea to just *import your flight plan* from the PS1 route file. You may have to dump the current PS1 FMC flight plan to disk first; either give that file a proper name or simply use 'ACARS' so that you can easily find it. PS1 route files can be found in the ROUTES directory in the main PS1 directory. Importing the FMC flight plan has more advantages than just ease of use. *Only when you import the flight plan,* SB747 *will be able to provide some route-dependant functions*! This influences mainly the external ACARS position reporting (which needs waypoint sequencing information) and the ability to gather forecast wind data for your flight route. These

functions won't work without an imported flight plan.

Submitting a flight plan also saves it in your preferences file, so the next time you start SB747, the flight plan will be there again. But it will *not automatically be submitted*, you will still need to do that manually by clicking the *Submit* button (after you reviewed the fields to see if they are still valid, of course). The *Clear* button erases all fields and gives you a blank sheet.

After you complete your flight plan and have submitted it (you can always resubmit it using the same button, replacing the old flight plan with the new one), you are ready to go!

3.3 Testing your SB747

First do a brief test to see whether SB747 can contact your PS1. Make sure the Broker is running and that SB747 has found it. This is announced by both the Broker and SB747. File a flight plan, no matter what, and click the three-way toggle to 'Robot'. SB747 should announce that the robot is engaged. Now switch to or start PS1, make sure the external data link is on (on the Preferences screen of PS1), and wait. After a short wait, a message should appear in the outside view window and you should hear some radio static (unless RogerWilco has been started, too; this will most likely lock out your PS1 sounds). Now hit 'K' and type 'hello' in the VHF transmit line that appears in PS1, followed by pressing 'Enter'. You should see your own message repeated in the outside view window. If this all works, you can be sure that SB747, the Broker, and PS1 are all set up correctly. Switch back to SB747 (with Alt-Tab) and switch the robot off.

3.4 Connecting to the Network

Make sure that you have a working Internet connection, for example by visiting your network's home page to check for special events and NOTAMs. Also make sure you have filed a valid flight plan. Then simply put the three-way switch on the SB747 main window to *Online*. The switch temporarily goes to standby mode, and after a few seconds it should settle in the *Online* position.

If it falls back to *Off*, the reason for the failure to connect is shown in the log window. There are various reasons for a failed connection, ranging from plain network problems to a call sign that is already in use. Correct the problem, then try again. After connection, you should get a few 'login' messages from the network, which should be noted but usually do not tell you anything new. You are now connected to the server cluster, with your flight plan already filed and you call sign operational. You exist in the virtual world.

The 'S' activity indicator should turn green and increase steadily. You can get a rough indication of the traffic in your area by watching this indicator; the quicker it increases, the more traffic. Should the indicator not run at all, there is most probably something wrong with either your connection or the server. Disconnect and reconnect again to see if that helps, or select

another server (in the Setup window). This does not destroy any of your settings or your flight plan. You can always just disconnect and reconnect later, but of course your plane will disappear in the virtual world for as long as you are offline.

As soon as you are online, you can minimize the running SB747, since you will not need to use it from now on. Everything can be done from the PS1 flight deck. In PS1, load in the situation you want, or set your position and other features yourself.

Congratulations, you can start flying the virtual skies!

3.5 Running the System

If you tune the radio to the correct frequency (look in the Controllers in Range window for nearby controllers and their frequencies), you will see and hear ATC messages coming by as usual, but this time they are live and interactive.

Go ahead! Tune into the clearance delivery or ground control frequency of your airport (or contact Tower, all depending on which controllers are available) and request clearance, using the terminology and phrasing you learned before. Soon you will be taxiing to the runway with quite some extra tasks to perform. Do not drive over those Cessnas 10 meters down below, please.

If ever something breaks and your network connection gets severed, you might need to cycle SB747 to *Off*, then back to *Online* to get the 'S' indicator running again. This will not affect your appearance in the virtual world except for a short hiccup. Every time you expect that something is broken, especially if the 'S' indicator stops running and turns red for more than 30 seconds, just cycle the connection. In case of doubt, *ping* a callsign that you know is present, such as 'server' who should always be there.

3.6 Quitting a Session

When you are done flying, first stop PS1, then switch SB747 to *Off*. The log will confirm your disconnect time. Now you use the `File/Exit` menu to quit the program, followed by a disconnection of your Internet line if you want to. The Broker should be killed last.

Chapter 4

Communications

This is what it is all about, of course. There can be no air traffic control without communications. In PS1, you have three communication decvices at your disposal to contact ATC: your set of VHF radios, your ACARS 'telex', and your transponder.

4.1 VHF Radios

Your main link to ATC is, of course, your radio set. PS1 has one tuning panel on the flight deck (in reality there are three) which you use like you have been doing it all along. The frequency tuned here is the frequency you receive and transmit on. You won't hear anything on other frequencies, except for the emergency frequency of 121.50 MHz which is automatically received all the time regardless of the tuned frequency.

In case you need to tune to a facility that uses a three-digit frequency, the two-digit PS1 tuner does not work. Use the **.freq** 'dot command' instead (Section 7.4). You can also use this command to review which frequency really is tuned, and to see which facility uses this frequency at the moment.

The system takes into account that VHF radio transmissions have a limited range and follow the line-of-sight rule. If you tune into New York while being over Europe, it will remain silent. While on the ground at Amsterdam Schiphol Airport, you will not hear London Heathrow Tower, but you might hear a plane enroute over London (if you tune the correct frequency).

4.1.1 Transmitting

You can transmit a message on the radio by pressing the 'K' key. Make sure the FMC keyboard is off (press Shift once to switch it on or off). A brown entry box will appear below your PFD, with space for 25 characters to type. Use only lowercase letters and numbers, and as few special characters as possible. PS1 has a very limited character set and you can easily fool its 'editor.' When your message is complete, press 'Enter' to transmit it on the tuned frequency.

The entry box is quite short, and more elaborate messages do not fit in completely. By appending a dot '.' to your message, SB747 is instructed to

buffer it and not transmit immediately. Each line entered that ends in a dot is added to the buffer. Only when a line not ending in a dot is typed, all of the buffer plus this last line is sent out. *Please make sure you give* PS1 *time to transmit each line to* SB747 by waiting a second in between pressing 'Enter' and 'K' for the next line.

An important character, the underscore '_' which is used in many ATC callsigns, cannot be entered in PS1. Instead you should use the dash '-' which is *automatically converted to an underscore* by SB747. Likewise, incoming underscores are converted to dashes so that you do not get confused (well, not so quickly).

In case you want to repeat your last message, there is a **.sa** 'Say Again' command available.

Lastly, there is an extensive *character expansion mechanism* in place that allows you to shorten messages that you use often to just a few keystrokes. More about this mechanism can be found in Section 7.5.

4.1.2 Receiving

Incoming VHF messages are displayed in the outside view window, in black on white. PS1 also tries to play a static radio noise sound with every incoming message. Should you not like this sound, it is in the `static.wav` file of the Broker's `lib` directory.[1]

In order to prevent your outside view to be cluttered up completely by a long message, SB747 breaks up long incoming messages into manageable parts. Each part counts as a separate message.

The 'Setup' dialog lets you specify the minimum time between received messages. You should find a balance between having enough time to read it all and not running behind too much. Four to five seconds usually turns out quite well. The Broker contains another setting, the time a message stays on the screen. This is essentially the time the *last* message stays on the screen, since incoming messages from SB747 will always replace the current message immediately.

When there is much chatter on the radio, and you have set the minimum time between messages on the high side, you will notice that some messages are followed by a trail of dots. Each dot represents a message that is waiting to be displayed. If this queue gets too long, you should decrease the minimum time between messages (in the Setup box), otherwise you will lag further and further behind ATC. Messages which contain your call sign are automatically put at the head of the queue, to improve responsiveness.

4.1.3 ATIS

Receiving ATIS messages is also a function of the radio set. However, the network usually does not transmit ATIS on VHF directly, neither with the "facility ATIS" bound to a callsign nor with the "AutoATIS" that is bound to a frequency. So in order to get ATIS you must send a special command, **.atis**. The reception of the ATIS will be channeled to your VHF radio set, though, and you will get the message displayed in the outside view window. In case

[1] Make sure that PS1 can get at this file, otherwise it obviously cannot play it.

you want to receive the ATIS associated with the facility you are already tuned into (either a facility ATIS or an AutoATIS), you can just type **.atis**. SB747 will try to find out the callsign by itself, or detect that you want a response from the ATIS server. In case you want to poll another facility, add its callsign to the **.atis** command.

Not all stations provide ATIS all the time, and once in a while the message format is less than desirable for PS1. In case of severe readability problems, temporarily switch back to the SB747 graphical user interface to peek at the log window.

4.2 ACARS

SB747 has pretty good ACARS support. It does not perform all ACARS functions exactly like in reality, because PS1 does not allow overriding of the ATIS and weather functions. But you have full access to the telex functionality, which can be used for about everything.

Important: SB747 has actually *two* ACARS systems. The one described here 'lives' on the virtual air traffic network and uses the MCDU of PS1. It is fully integrated in PS1, you do never need to leave the virtual flight deck to operate it. The other system, described in Chapter 8, uses a separate network that has nothing to do with either VATSIM or IVAO, and can only be accessed via the special MCDU that SB747 provides.

4.2.1 Telex

Incoming private messages ('chat mode' and system-wide broadcasts) end up in your PS1 flight deck MCDU. They will sound the EICAS chime and be announced on the scratch pad. You can use the standard ACARS TELEX page to review the message. If you have a message waiting to be read and another (new) message comes in, it will overwrite the waiting message. Each message is always preceded by the reception time in (PS1) Zulu.

By means of the ACARS TELEX SEND feature, you can compose a message to any other station or plane on the virtual ATC network. You must put the required callsign as the first word of the message, followed by the rest on as many lines as you need. After you press the SEND button of the FMC, the message will be sent directly to the facility. This can take a few seconds. If you try to send a message while the transmission of the previous message is still going on, the FMC will notify you with a 'buffer full' scratch pad message.

As a shortcut to reply to incoming messages using the sender's callsign as the address, you can use the **.r** command. See Section 7.4 for more information about all the 'dot commands.'

4.2.2 Weather Requests

You cannot use the native PS1 'Weather' functions with the flight deck MCDU, but a very close approximation is available through the **.metar** command. When you request a METAR from any airport, the network server will

look up a recent METAR for that station and relay it to your ACARS as a private telex message. As with all 'dot commands', you can put the command either in the VHF transmission line or in an ACARS telex. Using ACARS to request weather information is of course the most realistic.

4.3 Transponder

During virtual flights, the transponder becomes an important instrument. If you switch it off (called *squawk standby mode*, code 0000), you will be invisible to ATC, so you can drive around the tarmac looking for a place that suits you. Usually you are also invisible to other pilots when your squawk is on standby. As soon as you switch the transponder on (out of 0000), you will pop up on the radar screens and into other pilot's vision.

It is a good idea to keep your transponder off until ATC has assigned you a squawk. Note that in a real 744, the transponder is forced to standby mode whenever a wheel touches the ground, to protect sensitive equipment at the airport from being bombarded with high-energy transponder responses. It would be possible to model this in SB747 as well, but that would severely hamper the virtual ground controller in distinghuishing the planes on the ground. Most ground controllers do not have a visual 'tower' view, but use a ground radar screen only.

Some controllers do not anticipate the squawk standby mode, and will complain that they do not have you on radar after your initial clearance request. In such a case, please ask for a squawk code and put it into your transponder. In general, when a controller does not see you on radar, cycle the transponder first. Strangely enough this is *exactly* what happens in real life all to often, too!

You can send an IDENT signal with the **.ident** command. Do this only when a controller specifically asks for 'ident.' It will make your radar blip stand out like a sore thumb on his screen.

Chapter 5

Weather

When you switch on SB747, it will essentially generate no weather at all: no winds, no clouds, no nothing. In order to get some weather up there, you must specifically ask for it. This can be done manually and automatically.

5.1 Manual weather

Setting weather manually is done by the **.wxr** command. This command accepts the ICAO code of the airport you want to set the weather to. For example, if you are at New York J.F. Kennedy, you type **.wxr kjfk**. After a few seconds, SB747 will have set the PS1 weather to that reported for KJFK. It might take up to a minute before the PS1 'set weather' screen reflects the changes, but they are immediate. PS1 will take care that all changes are reasonably smoothed out – you won't find yourself suddenly in the middle of a roaring thunderstorm.

This mechanism supposes that you regularly reset the weather to that observed at certain airports. While enroute, switch on the ARPT feature of the navigation display and once in a while re-issue the **.wxr** command with an airport nearby. You need to get their METARs anyway for your enroute deviation airport checks. At cruise altitude you will not at all experience the low-level weather at those airports, so you do not need to reset weather all the time, but they will cause gradual upper wind changes, pressure changes, cloud cover changes, and temperature changes.

Together with the ground level observations, SB747 also sets the winds aloft to the values that the network relays to it. Although these values are not the observations in reality (unlike the METAR, the winds aloft are generated by a mostly random algorithm), they will be equal for all pilots in the vicinity. Only when they get really out of hand, you should use the **.winds** command to override the winds aloft.

In case you want to plan a long-haul flight, you can also use the **.winds** command to set the winds to the values you planned for. Since there are few airports in the oceans, it is probably best to use the last airport before the crossing as reference for the first half of the oceanic leg, and the first airport after the crossing for the second half. Or use the weather that is in effect at your current first choice deviation airport.

5.2 Automatic weather

If you switch on the weather updates (in the *Setup* box), SB747 will provide you automatically with the correct weather. The program determines where you are, and requests the correct weather for that area every five minutes. You can also force an immediate weather update to the nearest ICAO station by the **.wxr** command (without an ICAO code). Sometimes the network cannot provide the weather for the requested station. In that case, SB747 tries the next closest station, and continues until some weather is found.

SB747 shifts to a reduced set of weather stations when below FL240 to save CPU time. In cruise, all 6000+ ICAO stations are scanned for the nearest one. During departure or arrival, only the nearest 20 stations at the beginning of the flight or when crossing FL240 are used.

Network winds aloft are not always as physically logical as you would want. They might even behave counter-intuitive, e.g., with their direction decreasing with altitude on the Northern hemisphere. This is due to the incomplete weather model of most networks, it is not an SB747 issue. In case the winds are really annoying, switch off the weather updates and set your own winds with the **.winds** command and your own weather with the **.wxr** command.

5.3 Wind Aloft Forecasts

On long-haul flights, it is mandatory to have proper wind aloft forecasts available. Unfortunately, the online ATC networks do not provide real-world winds at altitude, so the normal skewed-T charts and other available resources are of no use for your flight planning. So SB747 had to contain a smart look-ahead feature, so that you can actually get the winds out ahead before you get there.

First, make sure you *imported a* PS1 *route file* in your flight plan, otherwise SB747 does not know where you want to fly. Then, go online, type **.probe**, and sit back. SB747 will fetch all the winds aloft on the waypoints of your flight plan for you, list them in the log window, and also dump them in the file `etc\winds.txt`. This feature is still under development, so it might produce slightly different information depending on what version of SB747 you are using.

During the probe sequence, your plane's radar echo will most likely blink a few times, so it is best (and most realistic) to probe the winds as part of your preflight and not while airborne.

We are aware that the current mapping from network winds to PS1 winds is less than optimal. Any good solution to map from the four network wind layers to PS1's continuous atmosphere model is welcome.

Chapter 6

RogerWilco

RogerWilco[1] is a free voice relay program that enables people to communicate with each other over the Internet, while running their favourite program in the foreground. You need a good headset (headphones plus microphone) and a supported full-duplex sound card in order to work with RogerWilco. Although the network will always support text communications, if only as a backup, many controllers are now 'on voice' using RogerWilco and it is quickly turning into the preferred mode of communication.

This manual does not detail the operation of RogerWilco, but some remarks about running RogerWilco together with PS1 are in order. First and foremost, you must usually start RogerWilco before PS1 since they might conflict in their use of the sound card. Running one often means the other cannot get the sound card any more. If you start RogerWilco first, it will grasp the sound card, and PS1 will complain that it cannot find the sound

[1] `http://rogerwilco.gamespy.com`

Figure 6.1: The RogerWilco Program

card. This is a pity, since you won't have PS1 sounds any longer, but if you want to use RogerWilco it is the only choice. Fortunately the latest sound cards such as the SoundBlaster Live! series (including the significantly cheaper Value Edition) support both PS1 continuous engine and event sounds *and* Windows (RogerWilco) sounds at the same time.

Second, most networks urgently ask you to use RogerWilco in PTT mode, i.e., you need to push a button to talk. The voice activation mode, which switches on as soon as it detects sound, is too much of a hassle to be practical. However, not all people can get RogerWilco to 'see' a key pressed in PS1. There are reports that the backslash key is suitable for use as a PTT key, but it is not guaranteed to work. In case you can only get voice activation mode to work, consider to put a physical switch in the microphone line to cut the audio feed manually. It works best to have the switch short the microphone signal, since this cuts out all hum as well. A regular light switch, meant for assembly in a lamp's mains wire, will do fine.

6.1 Tuning RogerWilco

The RogerWilco program must be 'tuned' to a facility in order to establish communications. A RogerWilco server (*host*) is a computer on the Internet, usually addressed by its IP number, such as 123.456.789.123. Several ATC facilities tend to share the same RogerWilco host, so on the same host there are several *channels*. You must know both the host and the channel of a facility if you want to contact to it. Most of the time, the correct host and channel are given in a station's ATIS, such as in 'Voice available on 193.130.4.110/EHAM_V_APP'. You can usually spot controllers who use voice communications because they have a _V_ in their callsign, but this is not a requirement. In the *Controllers in Range* box, you can get an immediate overview of controllers with a RogerWilco voice connection.

6.1.1 Manual tuning

The most direct way to tune RogerWilco is by manually setting the host and then manually connect to (join) the channel: **.rw h/193.130.4.110**, then **.rw j/eham-v-app**. The chance is high that you do not need to reset the host often as many facilities use the same host. Note also that you can (must) use the dash instead of the underscore.

When approaching an airport and directed by ATC to switch to another frequency, and therefore RogerWilco channel, having to retune the radio, listen in to the new station's ATIS, then retune RogerWilco is completely impractical. SquawkBox 747 therefore contains an Autotuner that facilitates RogerWilco channel switching.

6.1.2 Autotuning

Although not guaranteed fail-safe, the Autotuner will in most cases correctly find the appropriate RogerWilco host and channel for you.

Controllers are *required* to put their RogerWilco host and channel in their ATIS, so SB747 can look up this data for you. The program will ask

for the controller's ATIS as soon as the controller is recognized on the network, and re-fetch the ATIS every 30 minutes after that. Incoming ATIS messages are scanned for the RogerWilco host and channel, and if available, this information is stored in SB747 memory. You can review this list in the *Controllers in Range* box at all times.

You activate the Autotuner with **.rw a/on**. You only need to do this once, as this mode is retained in between SB747 sessions. In the rare occasion that you would want to deactivate the Autotuner, use **.rw a/off**. **.rw a** lists the current Autotuner mode.

While in Autotuner mode, SB747 will 'tune' RogerWilco *immediately* after you change VHF frequency – you have instant contact with the new controller. SB747 will tell you whether a RogerWilco channel was available, by flashing *Connected to XXX on voice* in the outside view window. Alternatively it tells you *Connected to XXX on text*.

If you suspect a station to have changed RogerWilco host or channel, you can re-poll the station for the new information with the **.rw p/callsign** command. If you leave the callsign out (**.rw p**), the currently tuned station is re-polled. This is very useful with many RogerWilco problems, as quite often a straightforward re-tune will solve them. In case SB747 announces *Connected on voice* and you don't get voice contact, try **.rw p**!

6.1.3 Future Developments

For true integration of RogerWilco, we need to wait until the networks complete a special RogerWilco mapping server, or integrates the RogerWilco channel in their server software. As soon as this facility becomes available, each controller will automatically get a RogerWilco channel assigned to her/him, and this channel can be looked up in a standard way over the Internet. SB747 will be adapted to this automatic lookup procedure as soon as possible after the facility is made available to the general public.

Chapter 7

Extra Features

The combo PS1 – SB747 – network has more features than 'just flying under ATC.' This chapter summarizes some of the extra features. You can do without them, but it is nice to know they are there just in case. There is no specific ordering in the presentation yet.

7.1 Lists of Pilots and Controllers in Range

Through the `File/Pilots in Range` menu, you can get a live list of the planes that are within your range, i.e., within approximately 40 nm from

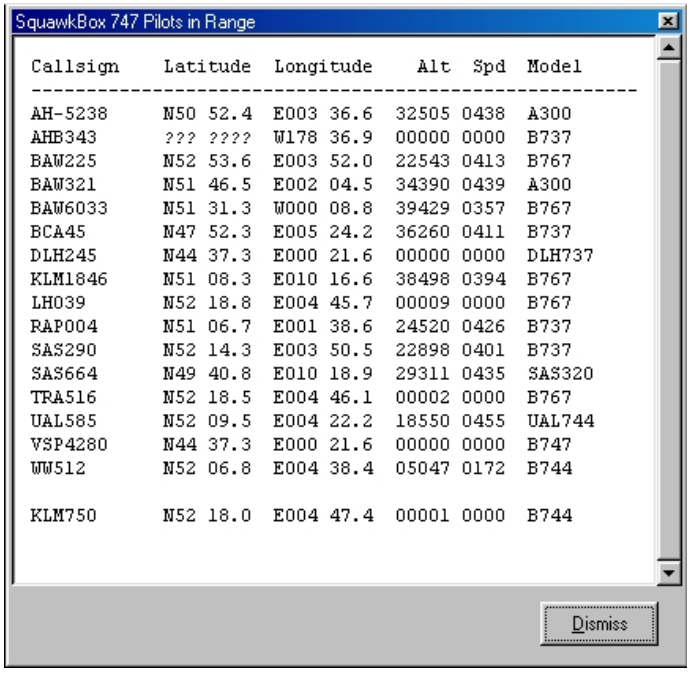

Figure 7.1: Pilots in Range Dialog Window

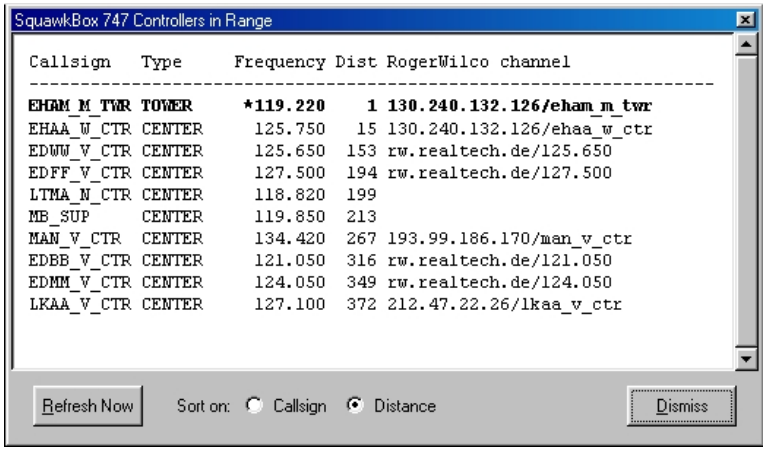

Figure 7.2: Controllers in Range Dialog Window

you. Network protocols do not deliver more than the plane's call sign, latitude, longitude, altitude, attitude (which is ignored by SB747), speed, and plane model. If you want to know more about a specific plane, such as the name of its pilot, you have to look at your network's Web pages.

On the last line of the list, separated from the others by an empty line, is the callsign and position of your own plane. This is mainly useful for checking whether you indeed are (in the virtual world) where you think you are. If this position is missing or consists of mainly zeroes, your PS1 connection has trouble. Check the Broker which data it shows.

A short form of about the same data can be displayed on the flight deck with the **.pilots** command. Also, the TCAS injector uses the planes on this list only.

The `Controllers in Range` menu analogously presents a list of Air Traffic Controllers in range. However, the definition of 'in range' is more specific here:

Ground Control	5 nm
Clearance Delivery	5 nm
Tower Control	30 nm
Departure Control	100 nm
Approach Control	100 nm
Center	400 nm
FSS	1500 nm

Generally you can communicate over the radio only with the controllers on this list, on the frequencies listed. The network Web page usually has a complete list of all controllers and pilots in the system.

The list shows the controller's callsign, the facility type, current frequency, distance to you in nautical miles, and its RogerWilco host and channel if available.

For your convenience, the facility that you are currently tuned to ap-

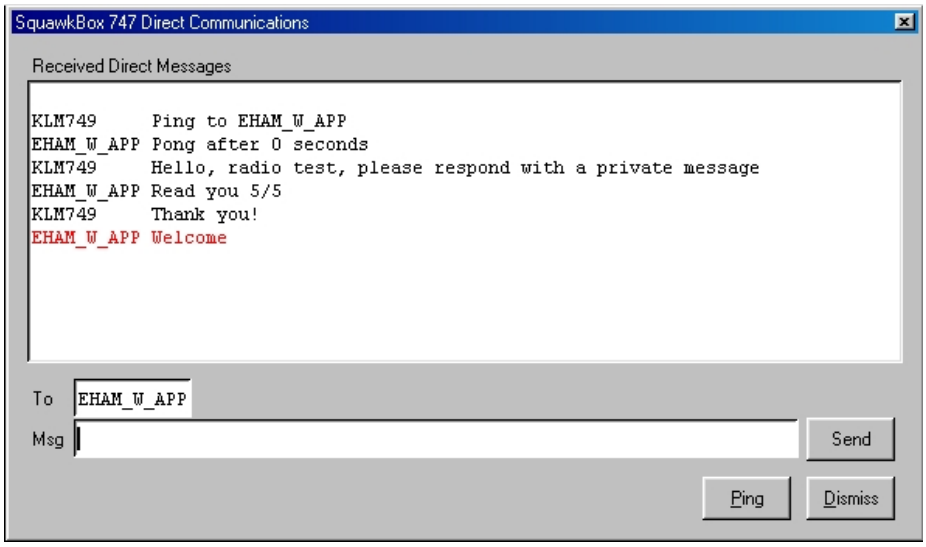

Figure 7.3: The Direct Communications Dialog Window

pears **in boldface**. The facility that is geographically nearest (give or take a few miles) has a "∗" sign prefixed to the frequency. This is not often useful, except when you appear in controlled airspace with no contact yet. You can then use the nearest controller plus common sense to decide which facility to contact first.

The list can be sorted on callsign or on distance to you.

The **.atc** command provides you with a shortened list of controller information on the flight deck, including the "+" sign for the nearest facility.

7.2 Direct Communications

Using the interface straight through PS1, you have only those options that a real airline pilot has to communicate: your transponder, ACARS, and a broadcasting transceiver. However, when you use the SB747 program itself (and thus temporarily suspend PS1 unless you have two computers), you have the option of comfortable *direct communication* with any other party logged on to the network. This is often useful, for example when polling an ATC what kind of services (s)he delivers (only approach, or also tower and ground, for example).

If you want to initiate a direct communication session, you open the `File/Direct Comm` dialog box, and fill in the receiver's call sign. The only difference between normal radio communication and direct communication is that you do not tune a broadcasting frequency, but you 'dial a call sign,' and communication therefore is private.

When somebody calls you directly while you are on the flight deck, you will get an ACARS telex message.

In SB747, you can also press the *Ping* button, which will send a ping request to the given call sign. The **.ping** dot command (Section 7.4) has the

same effect from the flight deck. When you receive the pong echo back, it will be displayed in the message window, together with the time that has elapsed between your ping and the pong echo. More than a few seconds lag time indicates a bad (slow) line somewhere between you and the ping target. You could try to switch servers to see whether that improves lag time.

As an extra feature, you can click on all lines in all windows (pilot list, ATC list, direct communication) where call signs appear. The call sign will then be automatically transfered to the 'To:' field. A double click opens the Direct Communications dialog box with this callsign.

7.3 Offline ATC: the Robot

In order to both save you money and provide you with ATC cover on long-haul flights through uncontrolled airspace, SB747 has an ATC *robot* that can take over from the human controller. The robot is designed to sound like it is actually controlling a sector, but the underlying program is simplified. The communications between the controller and the other (virtual) planes in the sector are realistic, though. You will have enough chatter over the radio to keep you alert for messages meant for you.

Currently, the robot is fairly simple, and it does not at all follow real-world sectors and restrictions. It even does not really listen to you, although it knows you are there and you should check on with Center after changing frequencies using the standard phraseology ("Center good afternoon, KLM 1534 with you level at flight level 370"). But since this part of the program is open to changes, it will grow in the future. It will become possible to add specific procedures for specific airports as well, which means that you then can use the robot not only for enroute (center) ATC, but also for departure and approach control, and even tower and ground.

For now, you must communicate with ATC through your flight plan. If you want another cruise altitude, or descend to the level where the PS1 ATC can take over (FL240), just alter your cruise altitude in the (SquawkBox) flight plan. The robot will get you there, eventually.

7.4 Dot Commands

From the flight deck, you can not only transmit radio messages, but also have a limited control of SB747. You have this control through the use of *dot commands*, words that start with a dot ' . ' on a new line. Available dot commands are (in alphabetical order):

. Empty command, causes the incoming VHF message window to be cleared. Useful in case some malfunction keeps your window cluttered with the last message received.

.atc Produces a list of controllers in range and their frequencies. Can be filtered, for example **.atc eh** produces only controllers whose callsigns start with 'EH'. Useful for peeking which controller you are likely to be switched to, and to find out which controller to contact initially.

The display order is the same one as the currently selected sort order in the *Controllers in Range* box.

.atis callsign Requests the ATIS of the given controller. Not all controllers have their ATIS installed. You should get a reply back in a few seconds, otherwise there is no ATIS. If you leave the callsign out, you will get the ATIS of the station your VHF radios are tuned into (if it has ATIS). Both facility ATIS and ATIS servers are supported; just tune in the right frequency.

.eta hhmm Updates the estimated time of arrival at destination. Needs to be done once in a while to let the external ACARS module know at what time you think you will arrive. You can copy the FMC estimated time from the PROGRESS page.

.fp Resubmits the flight plan, in case ATC asks for this. Should never be required but the network once in a while drops flight plans. If you type something after the **.fp**, it will be filed as remarks to ATC.

.freq Displays the currently tuned VHF frequency, including the callsign of the station currently operating on the frequency. If no callsign is displayed, the frequency is dead. When called with a parameter, tunes the VHF radio to the given frequency. This is useful for the new 8.33 kHz gridded ('channel') frequencies with three digits: **.freq 119.255** .

.help Gives all available dot commands.

.ident Causes your transponder to send an IDENT signal for five seconds. Use this only when the air traffic controller asks for it.

.metar icao-code Imitates an ACARS request for the METAR of the given airport. You will get back the most current true weather report of the airport as a real ACARS telex.

.msg callsign your message here Sends a direct communication message to the callsign given. This can be either a pilot or a controller. If you use 'r' as the callsign, you will **r**eply to the sender of the most currently received direct message. The same effect can be obtained with the **.r** command.

.network on,off switches on/offline. A flight plan must have been filed for this function to work, which can be done with the **.fp** command.

.pilots Like with the **.atc** command, you get back a list of callsigns and their current altitudes of the pilots (planes) in range. You can also filter this list with the first few characters of the callsigns you want to see, e.g., **.pilots klm**.

.ping callsign pings the given callsign. With no callsign given, pings the last station you received a private message from.

.probe fetches the current winds aloft of all waypoints in your filed flight plan, *if this flight plan has been imported from a* PS1 *route file*. The winds are shown in the log window and also saved in the `etc\winds.txt` file. They can be used for proper fuel planning and to feed the FMC more accurate wind information.

.r message here Replies to the most recently received private message.

.robot on,off switches the ATC robot on or off.

.rw Displays the currently set RogerWilco host (IP address). This is not necessarily the host the RogerWilco is now *tuned* to.

.rw a RogerWilco autotuning mode list.

.rw a/on,off RogerWilco autotuner on/off.

.rw h/hostIP Sets the RogerWilco host IP address for subsequent use by other **.rw** commands.

.rw j/channel Joins the channel on RogerWilco, using the host that was previously set by **.rw h/...**.

.rw l Leaves the currently joined RogerWilco channel.

.rw p Re-polls the currently tuned facility for RogerWilco info.

.rw p/callsign Re-polls the given facility for RogerWilco info.

.sa 'Say Again', repeats the last message you transmitted on VHF.

.scan on,off switches the radio to scanner mode. In scanner mode, *all* received communications are shown in the log window. This might be handy in some debugging situations, but will quickly fill up the window if you are in a crowded environment.

.winds Sets the winds aloft. A command without parameters deletes all winds. With a wind list given, PS1 will interpolate between the given winds to give the plane a wind changing with altitude and meeting the given wind points. A wind list looks like **.winds 120/340/12/20 240/010/25 370/030/40**. The first number is the flight level, followed by the wind direction and speed, and optionally the gust speed. You can give any number of altitude winds. However, the PS1 flight deck entry line can likely not even accomodate one, so you are advised to use the SB747 interface for this.

.wxr icao-code Fetches the METAR of the given airport (like the **.metar** command) and then sets the weather in PS1 to the received report. Also sets winds aloft to the network data, deleting the manually set winds with the **.winds** command. If you leave out the ICAO code, the command looks for the nearest ICAO station and tries to fetch the weather for that station. If that fails, it tries for the next nearest, etc.

7.5 Expanding VHF Messages

The `etc/expand.txt` file holds a list of phrase shortcuts or character expansions that you can use to minimise typing to ATC. The file contains entries like the following:

```
c   "$callsign"
aa  "$altitude feet"
clr "$org ground, $callsign at gate $1 request clearance for $dest"
ctr "$1 center, $callsign with you at FL$flightlevel"
cm  "climb to and maintain"
d   "direct"
dm  "descend to and maintain"
dyr "do you read?"
eil "established on ILS"
elo "established on localizer"
f   "FL$flightlevel"
fr  "out of FL$flightlevel for $1"
h   "heading $heading"
lvl "level at FL$flightlevel"
n   "negative"
ph  "present heading"
r   "roger, $callsign"
gs  "groundspeed $groundspeed knots"
tl  "turn left heading"
tr  "turn right heading"
u   "unable to comply"
w   "wilco, $callsign"
y   "affirmative"
```

Each 'word' in the left column is replaced by the phrase in the right column before the complete sentence is sent out. Note that there are several variables included in the expanded phrases; they all start with a `$` sign. These variables are filled in with actual values straight from PS1 at the moment of transmitting to ATC. They can save you some more typing, but are sometimes not completely accurate. For example, the `$altitude` variable produces your true height above sea level, as required by the network; not the pressure altitude according to standard atmospherical pressure.

There are also parameter variables such as `$1`. These are used when you write commands, like `clr`, that need extra information in order to generate a full sentence. If you type `clr/e6`, the expanded sentence will read '*KJFK ground, KLM123 at gate E6 request clearance for EHAM*' (or something similar).

You are encouraged to change the `expand.txt` file as you wish; the authors do not claim that the provided set of shortcuts is of any real practical value. For example, you may want to have each word that triggers an expansion start with a special character such as '=', to prevent accidental triggering of expansion rules.

7.6 Ground Position Tracking Tricks

Since PS1 is an IFR simulator, ground scenery is sparse at best. It is nearly impossible to follow taxiways, even if you have a detailed map at hand, because the outside visuals are not more than Spartan outlines of runways

plus some mysterious light poles. No gates, buildings, taxiways, or holding lines.

To solve this problem, a separate program that integrates into PS1 has been developed. You can download it from the Internet.[1] This 'Taxiway' program displays the appropriate ground map, straight from the controller's screen, on your lower EICAS. An extension, called TF, is able to display the actual network traffic as TCAS symbols on top of this map, giving you a good overview of traffic in the neighborhoud (up to about 640 nm distance).

Alternatively you can use the position of airport gates[2] to create a 'waypoint' on the navigation display, so that you have some idea where you are.

[1] http://www.geocities.com/ps1ipc/twy.html
[2] http://infolab.kub.nl/people/hoppie/ps1/gates

Chapter 8

External Systems

Next to the 'internal' chat functionality of the PS1 flight deck CDU, as discussed in Section 4.2, SB747 offers a completely separate implementation that goes far further in capabilities. This system contains currently both an advanced ACARS system and a complete interface to the online ATC systems. In principle, you can use this interface instead of the native Windows SB747 interface during most if not all of your flight. This provides a much better look and feel than the point-and-click interface which seldomly is encountered in a real airplane.

8.1 The external MCDU

You can get at the extended ACARS functionality only by installing the *external* MCDU, a separate program available from the same web site as where you downloaded SB747 itself. This program displays a simplified MCDU on any computer you attach to your network. You will quickly feel at home with it. All required buttons and keys work, and you can use either the mouse or the standard PS1 MCDU keys to operate it. We had to program our own MCDU in order to implement more advanced ACARS features beyond a simple one-page telex.

If you have the external MCDU attached, PS1 will stop using the internal PS1 CDU for all purposes except to tell you that a message appeared on the external MCDU. However, you are still free to use PS1's CDU to send telexes or other things.

You are encouraged to find creative ways of encapsulating the external MCDU in your virtual cockpit; the program has plenty of features to blend in seamlessly in most hardware. You can even start more than one copy and have them all operate in a different mode, or in parallel in the same mode. See the MCDU manual for more information.

8.2 Basic ACARS principles

The SB747 ACARS module is based on store-and-forward of discrete messages. Imagine a network of ground stations (typically airline dispatch

offices), one central communication station, and many airborne stations. Each station has a unique call sign, and you can send messages to other stations by addressing them to the correct call sign. Each message travels to the central communication station, where it is stored. When the addressee station checks in, the message gets delivered.

You can have a peek at the central communication station on the Web.[1] A separate application called ACARS747 is available to take on the role of ground station. Neither of these systems are exclusively for PS1, so if you want to develop your own ACARS add-on, feel free to build further upon the common infrastructure.

Essential in the system is that all stations drop by the communication station regularly to 'pick up mail.' If they don't, their mail is destroyed 24 hours after being sent. So if you do not get any reply, your addressee might simply not be online (yet). An exception is made for pre-stored messages that are sent by a station to the communication station as some kind of 'poste restante:' you can ask for these messages at any time, and they will get delivered to you instantly. Typical examples of these drop-off messages are load sheets, arrival information, and (live) weather bulletins.

ACARS only works properly if you have a near-constant Internet connection during your flight. This is why the package is part of SB747, which already assumes such a connection. However, the ACARS system is completely independent of the online network of your choice, with the exception of the weather (METAR) retrieval functions.

[1] `http://infolab.kub.nl/people/hoppie/ps1/fl/acars/`

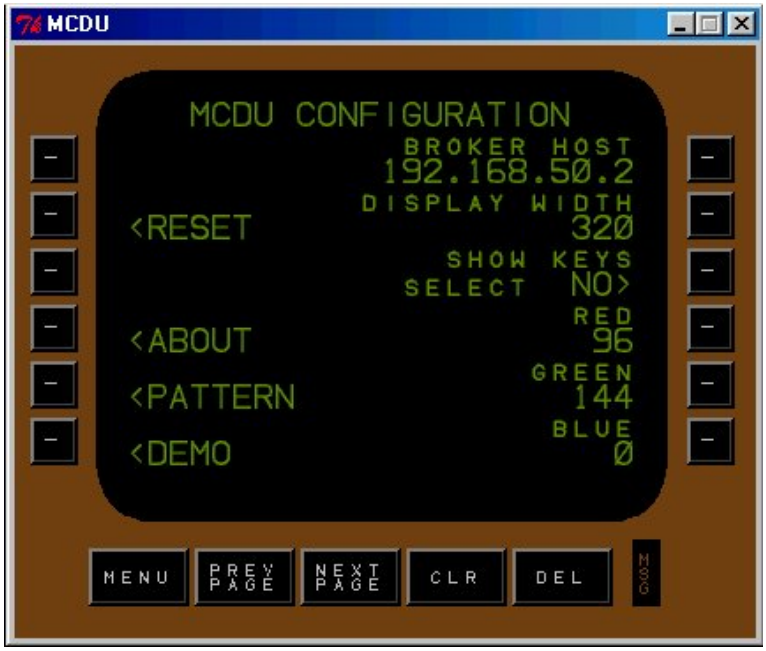

Figure 8.1: The External MCDU Window

8.3 ACARS menu structure

The menu system of the ACARS module is not a perfect copy of a real airline ACARS system, because it is very hard to get proper information and every airline seems to have a completely different setup. If anything, it resembles the Qantas model mixed up with Lufthansa. However, the functionality is largely the same over airlines.

ACARS has three basic modes: preflight, inflight, and postflight. When you call up the ACARS menu, it will automatically drop into the most appropriate mode, but via the ACARS *index* choice you can always go to the main menu. This main menu also offers the *Link Status* choice, with which you can check whether your (Internet) ACARS link is operational. If not, the system will tell you why not.

8.3.1 Preflight

Here you set up most of the required data elements. During flight, there is much less to do. If you file a SB747 flight plan to the ATC network, many of the details on the *Preflight* page will be filled in for you automatically.

The *System Init* prompt causes a cleanout of the ACARS system, which should be done before each flight (or, as a courtesy for the next pilot, after the flight). You need to push the LSK1 two times.

Origin, *Destination*, and *Alternate* fields should contain the ICAO codes of your departure and arrival airports, plus the most likely alternate. These fields will be used throughout the ACARS module for reports and requests. They are not linked to your FMC.

In the right column, you can fill in your flight details, such as *Flight number*, *Planned Departure Time*, and *Estimated Time of Arrival*. The latter will be automatically recalculated when ACARS detects that your plane moves out of the gate, by adding current time and estimated flight time from the flight plan. You can manually reset it at any time. Currently the ETA is not yet taken over from the PS1 *Progress* FMC page.

The flight number is used as your ACARS callsign.

Figure 8.2: The Preflight Page

Figure 8.3: The Inflight Page

Very important is the *Company* field. Here you fill in the three-letter acronym of your company, such as KLM, CLX, or STV. If you fly for a virtual airline, your airline will tell you the company code to use. Most of the time, this is the address on the messages you send, but you can override it in some cases (these will be explained).

In principle, you will want to have all fields on the *Preflight* page filled in before departure.

8.3.2 Inflight

After you leave the gate, or by manual request, the *Inflight* page appears. This contains the items that you might need during flight, up to the moment you dock at the arrival gate and shut down.

Since most ACARS systems work automatically, you do not need to do much here. If an external station requests a position report, it will be sent automatically, but you can forcefully send one via the *Position Report* choice. This opens a page where you cannot change anything, since it is all collected automatically. For the correct waypoint tracking, you will need to have imported the flight plan from a PS1 route file in SB747. Some items on this page are inoperative due to the limited access that SB747 has to the PS1 FMC; this might improve in the future. A manual transmission of the position report is always addressed to your company ground station.

The ETA can be manually updated. Large deviations might trigger an automatic progress report to your home base. You should keep this figure more or less synchronised with the PS1 FMC *Progress* page, which in the future might be done automatically.

In case you decide to deviate to the prefiled alternate, you can simply press the LSK2L button. This will overwrite the destination with the alternate and ask for a new ETA. You can also put a completely new ICAO code in this field, in case you want to deviate to a different field, or continue back to your original destination. Deviations are automatically reported to your home base.

Figure 8.4: The Event Times Page

8.3.3 Postflight

You end up here after docking at the arrival gate, or by manual selection. About the only option here is the *Event Times* page, which summarizes important operational data of your whole flight. This page contains the data that are included in the 'OOOI' (out-off-on-in) progress reports which ACARS sends to your home base as soon as something important happens:

Pushback Out time and ETA are sent.

Takeoff Out and off times, and ETA are sent.

Landing Out, off, and on times are sent.

Docking Out, off, on, and in times are sent. Docking time reporting is delayed until the plane has been stationary for more than two minutes.

ETA changed more than ten minutes and plane is in last 120 minutes of flight Out and off times, plus revised ETA are sent.

20 minutes before ETA Out, off, and ETA are sent.

You cannot change these data, as they are automatically acquired from several airplane systems and cheating is not approved by your company.

Out Time you leave the gate; first motion of the plane after ACARS initialization.

Off Time you get airborne.

On Time you touch down.

In Time you stop the plane for the last time after landing (the system waits two minutes to be sure you really won't start moving again).

ACARS calculates block time and flight time for you, so you can copy these figures into your flight log. As OOOI progress reports are sent to your company automatically, you do not need to manually send these administratively interesting times.

8.3.4 Received Messages

The *Received Messages* selection brings you to the message log of the external ACARS facility. Here you can access all messages that the system received from a ground station (uplink). There is no downlink log.

The page opens up a list of all messages received, and you can select which one you want to view. A maximum of nine pages of five messages each can be retained by the system. If it fills up, the oldest messages go overboard. Each message can hold nine pages of five lines of 24 characters. Message lines in large font are not yet reviewed by the crew ('new').

If you receive a new message while not viewing the *Received Messages* page, the MCDU scratch pad will pop up a message ACARS UPLINK to warn you (and the PS1 CDU will warn you, too).

8.3.5 Requests

Various requests can be made by the crew via ACARS. They are all collected on one single page, accessible from many places in the menu system.

Route You can download a complete FMC route, so that you do not need to leave the cockpit at all for file handling, and also do not need to key in lengthy sequences of boring data. Your company must have made the route available to you, and tell you the company route code. If you have it, you may enter the code and press *Send*. If the route is available, it will be downloaded into your FMC and you may pick it up as follows:

1. Go to the PS1 FMC RTE page.
2. Fill in ACARS in the *co route* field.
3. Review, activate, and execute.

Route repositories are company-specific. For more information about setting up a route repository, contact the author. **A note**: SB747 saves the route in the same directory as where you imported your last PS1 flight plan from. You might need to import one flight plan once in order to set this directory (use the SB747 File/Flight Plan menu, then click on Import).

Weather This only works when you fly online on one of the ATC networks. You can select one of the three known ICAO codes (origin, destination, and alternate airports) or key in any other station you want to get weather from. When you press *Send*, the large font ICAO is sent out for weather information. The resulting coded METAR is sent to your generic messages box, where you can keep it for later review. This also means that you may request more METARs in a sequence; they will all be stored as they come in.

Load sheet A very airline-specific item. If your airline uses this feature, you may pick up the final load manifest of your plane and drop it in your received messages buffer.

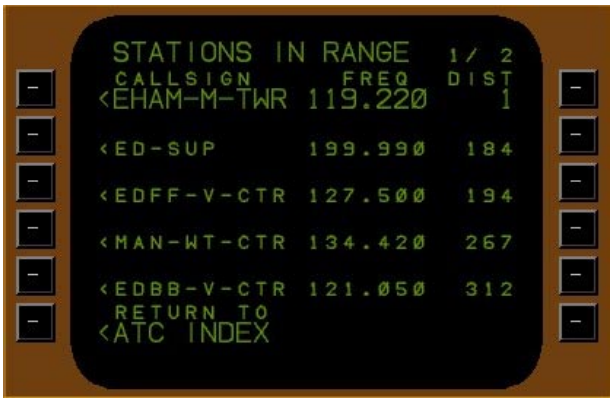

Figure 8.5: The ATC Page

Arrival info Another airline-specifc item, used for advance information about your destination: gate assignment, maintenance expected, crew rotation, connecting flights...

Send Telex Used for free-form messages to home base or any other ACARS station you happen to know the callsign of. Remember that messages are stored in the central communication station until they are picked up, which might not be immediate if your addressee is not online yet.

If, for one reason or another, the transmitting of a message fails, the *Send* button will change to *Resend* and you may try again. The *Link Status* choice in the main ACARS menu offers more detail if something seems to go wrong.

8.4 ATC functions

On the external MCDU, you will also find a live list of ATC stations 'in range' and a complete radio tuner, including RogerWilco (re)tuning. These pages are obviously not 'realistic' but can be very handy if you wonder what is going on online. They enable cockpit operation of otherwise very unnatural machinery.

On the *Stations in Range* page you see all stations currently within radio range, ordered as you have ordered the same list in the Windows SB747 interface. You can select one of the stations by the push of a button, which will instantly re-tune PS1's radios. In some pathological cases, this works better than actually using the real radio tuners; however, it is a hack. The station you are currently tuned to appears in large font, the rest in small font.

The *Radio Tuning* page shows the currently tuned frequency, the station's callsign if there is a station operative on this frequency, and its RogerWilco details if available. You may re-poll the station for RogerWilco retails with LSK6R. Once in a while this saves the day. Otherwise you may manually override all items on this page except for the callsign.

Figure 8.6: The Radio Tuning Page

Figure 8.7: The GoFlight GF166 Versatile Radio Panel

8.5 GoFlight Hardware

GoFlight Inc.[2] produces several hardware modules that have been integrated in SB747. You can tune the PS1 radios with it, and select transponder codes. For more information, visit my web site.[3]

- o - o - o - o -

[2] http://www.goflightinc.com/
[3] http://www.hoppie.nl/gfsb747/